testing, upon activation of said exit collection, other members of said exit collection for said member's active/inactive state and if any member of said exit collection is inactive, then stop testing and return said exit collection to its inactive state, and otherwise, if all members have tested active, activate their common exit node.

REMARKS

This is a refiling of the amendment filed May 29, 2001 to comply with 37 C.F.R. §1.121. The remainder of the amendment remains the same except for a modification of new claim 11 and the correction of several typographical errors, in claim 10, and in the remarks.

Reexamination and reconsideration of the claims, as amended, are respectfully requested.

The Examiner (a) has rejected claims 2–7 as being directed to non-statutory subject matter, (b) has objected to claim 5 under 35 U.S.C. § 112, and (c) has rejected all of the claims under 35 U.S.C. § 102(e) as being anticipated by Nock, U.S. Patent 6,144,967. The Examiner has also objected to various elements of the specification and drawings. For the reasons noted below it is respectfully submitted that the claims, as amended, are patentable over the Nock reference, are directed to patentable subject matter, and meet the requirements of 35 U.S.C. § 101.

The drawings will be amended, as noted in red on the attached drawing sheets (these are the substitute sheets of the priority PCT application), to meet the Examiner's objections. Subject to the approval of the Examiner, formal correction will be made. The description has been amended; and in addition, a separate page of abstract is being enclosed,

which is the same as the abstract appearing in the PCT application from which this application obtains priority. In particular, reference numbers have either been added or deleted in various sections of the application or drawings.

With regard to the non-statutory subject matter rejection of claims 2-7 under 35 U.S.C. § 101, it is respectfully submitted that the application, as amended, fully meets the requirements of the statute, in particular, in light of the new Patent Office Examination Guidelines for Computer-related Inventions. It should be clear from the specification that the invention is created as a computer-based system for implementing methods and systems as claimed. In this respect, the claims have been clarified to more specifically refer to a computer-based system and method.

Nevertheless, a review of the application as filed is enlightening. The title relates to a "computer-based system...". The abstract relates to a "computer-based method and apparatus...". The field of the invention states that it is "the field of computer-supported collaborative work." It specifically further calls for "(4) implementing computer-based systems to support...such collaborative work." The background relates to workgroup software, work-flow software, and early decision support software. The summary of the invention states, at page 8, with regard to the models of the invention, that "[t]hese project models are a central element in a computer- and communications-based infrastructure to direct and guide the behavior of the participants in the work process". The summary further relates to "workflow and workgroup software, information technology", and so on.

Because of the nature of the implementation, that is object oriented programming, the description does not fall into the generally viewed form of a typical procedural computer program. Nevertheless, the description clearly describes the objects which are to be used in

connection with the invention and the classes of those objects as those terms are used and well known in connection with object-oriented programming. Further, with regard to, for example, the functional model, the description states "a data flow diagram represents a computation" (page 17, lines 16-17) and that "[o]bject-oriented development places a greater emphasis on data structure and a lessor emphasis on procedural structure than traditional functional-decomposition technologies" in quoting (at page 17, lines 25-27) from the definition of object-oriented modeling and design, referred to on page 16 beginning at lines 15-18. Indeed, the Object Modeling Technique (OMT) is here a computer-based system which is being described.

Further, the application at page 18 describes the term "application framework" as a "set of abstract and concrete classes comprises a generic software system for an application domain". At the bottom of page 18, the text further states, beginning beginning at line 27, "[t]he present invention consists of an "application framework" for development of abstract, decision process models." Note that this is the same "application framework" defined as a generic software system for an application domain. Further examples are found in the text, and will not be further specified. Nevertheless, it is respectfully submitted that the amended claims and the description clearly provide sufficient basis for withdrawing the rejection based upon non-statutory subject matter.

In addition, Applicant has added additional claims to more fully and clearly claim the scope of the invention to which he is entitled.

Further, claim 5 has been rejected under 35 U.S.C. § 112. The Examiner has objected to the language "utilizing messaging". It is unclear why the Examiner views this language as indefinite. The specification beginning at page 29, line 10, begins a description of

the dynamic behavior of directed arcs, arc entry collection objects, and arc exit collection objects, and in that description clearly refers to the use of messaging or notifications between various states of the various objects in this decisional process. That is, once instantiation occurs, the method in fact utilizes the sending of messages (messaging) between the nodes and arcs, and the collections of arcs. Accordingly, it is suggested that the language of claim 5, "utilizing messaging" is clear. (Note that here, "messaging" is used as a noun.)

All of the claims have been rejected under 35 U.S.C. § 102(e) as being anticipated by Nock, U.S. Patent 6,144,967. Nock, however, does not meet the terms of the claims for the following reasons.

The subject matter of Nock is very "loosely" related to that of the claimed invention. Nock's framework produces object models of highly automated processes for analyzing computer logs or files. In its preferred embodiment, it has only one participant, the operator, who makes only a couple of decisions. This is quite different than the object-oriented multi-participant system of the claimed invention which produces object models of processes which can range from completely manual (other than the communication and control which the model's implementation automates using a computer system) to a fully automated system. Unlike Nock, in the claimed invention, the data need not be restricted to computer logs and files. Furthermore, the processes to which the claimed invention is directed are those that are fundamentally collaborative requiring interactions among participants. This is why in Konnersman there are classes of decision, decision roles, and data which are important to the invention. In Nock they simply do not even exist.

With regard to claim 1, Nock does not address processes or projects but deals explicitly with one process namely that of analyzing a computer generated log. This process differs substantially from those which are the objects of the claimed invention, which anticipates the need for multiple interactions between a computer-based system and numerous participants. Thus, claim 1 requires both plural process models and plural project models and making those models elements of a computer-based system. In Nock, to the contrary, there is only one specific process which is dealt with, not a number of process models and project models. Accordingly, Nock simply does not describe, suggest, or teach the claimed invention. Further, Nock merely takes a procedural program and implements it in object form; this is not possible with the claimed invention.

With regard to independent claim 2, there is, in the elements referred to by the Examiner, nothing which relates the decision object to one or more data objects, nothing which "requires" for at least one decision object at least one data object as a prerequisite to its activation or completion, nothing which optionally generates additional subclasses or instances of the decision and data classes, nothing which instantiates a plurality of objects by abstract or concrete classes and nothing which relates each decision object to one or more data objects which it produces. The structure of this method claim is not met by Nock and the operation of Nock is merely that of a typical object-oriented system, and is unlike that of the claimed invention which relates back, clearly, to the description herein.

Nock further does not provide for messaging between the nodes and arcs, and the collections of arcs as called for and described in the claimed invention (for example, claim 5). Nock does not provide any such structure between his objects even though, of course, as in any object-oriented system, he does identify and provide for communications between the objects.

The interrelationships called for by claim 5 and the communications therebetween are simply not described in, for example, Figure 5 of Nock. In addition, claim 6 which requires a network with nodes as abstract decision situations and arcs directed by decisions based on logical precedence, is not described anywhere within Nock. In particular, it is not described in Figure 5 wherein one has merely the ordinary flow of information in an object-oriented program. The management of work processes is contemplated by claim 6 by providing arcs directed by decisions based on logical precedence is not found in that Figure. Similarly, the Figure does not describe a method for modeling and managing work processes as called for by the claim. The remaining original claims are dependent upon the claims noted above and should be found patentable at least for the reasons that the claims upon which they depend are patentable.

Applicants have also added claims 8-13 to more fully cover the invention described herein. Those claims have been reviewed in light of the Nock reference and are considered to be patentable over that reference, as well as the other references of which we are aware. Applicants are further submitting, under separate cover, an information disclosure statement which will cite references found during the PCT prosecution of this application as well as several other references of which applicant is aware. It is respectfully submitted that none of those reference adversely affect patentability of the claimed invention.

Further, Applicant's attorney has reviewed the other references cited by the Examiner and does not find them to adversely affect patentability of the claimed invention. Accordingly, Applicant respectfully requests that the claims, as amended, be passed to issue in due course, in view of the comments and amendments made herein.

The Commissioner is authorized to debit any necessary fee or credit any overpayment relating to the above-identified application to Deposit Account No. 08-0219.

Respectfully submitted,

Date: 9/24/01

Gary A. Walpert
Reg. No. 26,098

Hale and Dorr LLP
60 State Street
Boston, MA 02109
Tel. (617) 526-6000
Fax (617) 526-5000

Marked up Version of Replacement Paragraphs in Specification Under 37 C.F.R. § 1.121 (b)(1)(ii)

On page 3, please delete the paragraph beginning at line 22, and replace it with the following paragraph:

Each data element, whether elementary, a [sub] sub-assembly or final assembly, is the product of a decision. That is, it is selected from two or more alternatives. For example, the color specified in a product specification might be red, green, or blue. The business plan that includes a $3 million advertising budget could have instead, included one for $2.5 or $2 million. Or it could have included separate line items for advertising by geographic region, or by type of media, or by product line, or various combinations of these possibilities. What it does contain is a matter of choice (i.e., a decision) and results in data.

On page 6, please delete the paragraph beginning at line 6, and replace it with the following paragraph:

In the '80's "participative management" was superseded by the coming of "teams" with similar results. It is often at least useful, and perhaps essential, for a group of people to work together interactively-as a team. It is seldom adequate however, to roundup a group, anoint them with "teamhood," provide team T-shirts and send them off to play the game. The football, basketball, and other teams that provide the model for organizational teams, don't usually play the game without

considerable investment in learning how to "block and tackle" and *then* practicing the "blocking and tackling" repeatedly until they do it *really* well. They also develop "play books" and shared understanding of cryptic [signals,] signals. They learn to anticipate each others moves, again as a result of much practice together. Where are the organizational equivalents of these indispensable requirements for the success of athletic teams? What one usually finds is a one day, or at most one week course, followed by a return to the workplace bearing an appropriately emblazoned coffee mug and plaque for the wall.

On page 7, please delete the paragraph beginning at line 9, and replace it with the following paragraph:

We need a method to analyze, specify and support work processes that consist of many, interdependent decisions, at least some of which require collaboration among multiple participants for satisfactory results. This is at least part of an answer to two critical problems currently faced by most complex organizations -1) How to get better integration of effort across organizational boundaries, both those created within organizations (e.g., between engineering and manufacturing, or eastern sales region and central sales region) and the boundaries between organizations (e.g., customer and supplier, business and government, federal government and state government), and 2) How to improve the performance of managerial and professional work, where such performance may be measured in terms of reliability of the process in producing quality output, the productivity of the

process, or the speed of the [process] process.

On page 16, please delete the paragraph beginning at line 11, and replace it with the following paragraph:

A *class* is an abstraction that describes properties important to an application and ignores the rest. ... Each class describes a possibly infinite set of individual objects. Each object is said to be an *instance* of its class. (James Raumbaugh, Michael Blaha, William [premerlani] Premerlani, Frederick Eddy, and William Lorensen, *Object-Oriented Modeling and Design*, Prentice Hall: Englewood Cliffs, NJ, 1991, p. 2)

On page 17, please delete the paragraph beginning at line 8, and replace it with the following paragraph:

The *dynamic model* describes the aspects of a system that change over time. The dynamic model is used to specify and implement the *control* aspects of a system. The dynamic model contains state diagrams. A *state diagram* is a graph whose nodes are *states* and whose arcs [arc] are *transitions* between states caused by *events*. ...

On page 19, please delete the paragraph beginning at line 13, and replace it with the following paragraph:

The Framework 99 is constructed from a related set of abstract and concrete object classes that are depicted in FIG. 6. The abstract Decision class 100 has members that are classes of decisions which are specific to the application

domain. In the example, depicted in FIG. 4, all of the boxes representing nodes of the network would be modeled as concrete class instances of the Decision class 100. This relationship between the abstract Decision class 100 and some of [it' s] its concrete classes and object instances are more clearly depicted in the upper half of FIG. 6. The Data class 101 is also an abstract class that has a one-to-one relationship with the Decision class 100. The relationship between the abstract Data class 101, its concrete classes and their object instances is shown in the lower half of FIG. 6. Referring again to FIG. 5, the other abstract classes of the Framework 99 are Arc Collection 115 and Decision Role 121. The Arc Collection class 115 has two concrete subclasses, Arc Entry Collection 134 and Arc Exit Collection 136. The instances of these classes are collections of Directed Arc 107 objects which are instances of another one of the Framework's 99 classes. These two subclasses are differentiated by the end of the Directed Arc 107 object that they use to determine their members; the former using the entry end of the Directed Arc 107 object (the end without the arrowhead in FIG. 4) and the latter using the exit end. The abstract Decision Role class 121 has five concrete classes in the preferred implementation, Decision Manager 142, Consultee 143, Approver 144, Inspector 145, and Informee 146. These [four] five concrete, subclasses model the behaviors and

responsibilities described in Table A. As indicated in FIG. 5, there will be exactly one Decision Manager 142 related to each Decision 100. There may or may not be any Position 119 designated to participate [120] in a Decision 100 in any of the other four roles 143, 144, 145, and 146. Nor is there a limit on the number of Positions 119 that may participate [120] in any of these latter four roles. The final classes of the Framework 99 are the concrete classes Position 119 and Person 116 which model the organization and the incumbents of the organization respectively.

On page 20, please delete the paragraph beginning at line 22, and replace it with the following paragraph:

The Framework **99** depicted in **FIG. 5** has both abstract and concrete classes but no objects. Two of its classes do not have any concrete classes. **FIG. 7** depicts classes and objects of a hypothetical Process Model **129** derived from the Framework and based on the example depicted in **FIG. 4**. In addition to the elements of the Framework depicted in **FIG. 5**, the Process Model **129** has concrete subclasses Cost **10**, Price **11**, Terms **12** etc. of the of the abstract Data class **101**, and concrete subclasses Cost ? **14**, Price ? **15**, Terms ? **16** etc. of the of the abstract Decision class **100**. (the short broken lines **13** and **17** indicate that there are other concrete subclasses of these two abstract classes which have been omitted for clarity.) The Framework **99** abstracts the desired behavior common to all decision processes whether

they be a proposal preparation process, a product development process, or a strategic planning process. The Process Model **129** is more concrete and specific. It [abstract] abstracts only those desired behaviors that are common to the particular decision process being modeled, in the example illustrated in **FIG. 4**, **FIG. 6**, and **FIG. 7**, the proposal preparation process of the organization or organizations that use this particular process. The Process Model **129** also includes the objects which are instances of the concrete classes Directed Arc **107**, Arc Entry Collection **134**, Arc Exit Collection **136**, Position **119**, and the five concrete subclasses of the Decision Role class **121** to the extent that any are specified for this particular process.

On page 22, please delete the paragraph beginning at line 20, and replace it with the following paragraph:

Requiring **105** Decisions **100** and their dependencies upon producing **110** Decisions **100** are connected by Directed Arcs **107** with an entry at the end of the arc connected to its respective producing **110** Decision **100** and an exit at the end of the arc connected to its requiring **105** Decision **100**. Each Directed Arc **107** [ia] is a member **133** of one Arc Entry Collection **134** comprised of **133** all and only those Directed Arcs **107** which have the same producing Decision **110**. Each Directed Arc **107** [ia] is also a member **135** of one Arc Exit Collection **136** comprised of **135** all and only those Directed Arcs **107** which have the same requiring Decision **105**. Arc Entry Collections **134** and Arc Exit Collections **136** are specializations of the Arc Collection **115**

class, which specialization is based on whether the class is defined by its entry **109** relationship or its exit **108** relationship.

On page 23, please delete the paragraph beginning at line 27, and replace it with the following paragraph:

(1) Decision Role **121** applicability **125**: "IF {product category] = {lawn care}, THEN {Decision [Manger] Manager} = {Product Manager, Lawn Care}, ELSE IF {product category} = {snow blowers}, THEN {Decision [Manger] Manager} = {Product Manager, Snow Handling}, ELSE {Decision [Manger] Manager} = {Marketing Manager};"

Marked up Version of the Amended Claims Under 37 C.F.R. § 1.121(c)(1)(ii)

1. (Amended) A method for instantiating project models as instances of a process model to which they conform comprising [the steps of]

supporting the work [in] of the process by rendering said process models as elements of a computer-based system, and

supporting the work of the process by rendering said project models as elements of a computer-based system.

2. (Amended) A computer implemented method for modeling work processes comprising [the steps of]

instantiating a plurality of objects by abstract or concrete classes, and including at least a decision class and a data class,

relating each decision object to one or more data objects which it produces,

requiring, for at least one decision object, at least one data object as a prerequisite to its activation or completion, and

optionally generating additional subclasses or instances of said decision and data classes.

3. (Amended) The method of claim 2 further comprising [the step of] relating an arc or link class linking a first decision with a second decision.

4. (Amended) The method of claim 2 further comprising [the steps of]

generating a decision role class specialized into at least two subclasses, each with differing behaviors, and

defining for each decision role class the communication requirements among the incumbents of roles participating in a decision, the rights of each such role class incumbents with respect to (a) entering data elements in a database, (b) modifying elements in a database and/or (c) reading elements from a database.

5. (Amended) A computer implemented method for traversing networks including nodes and directed arcs comprising [the steps of]

utilizing messaging between said nodes and arcs and collections of said arcs, and

determining the membership of said collections by at least one of their entry nodes and exit nodes.

6. (Amended) A computer implemented method of modelling and managing work processes among a plurality of participants comprising [the steps of]

using a network whose nodes are abstract decision situations, and

providing arcs directed by decisions based on logical precedence.

7. (Amended) The method of claim 6 further comprising [the step of]

requiring nodes to support participation of multiple persons in differentiated roles.

Add Claims 8-13 as follows:

8. The method of claim 7, further comprising

requiring that incumbents of exactly one differentiated role make a choice modeled by an abstract decision situation, and

requiring that the incumbents of a second differentiated role have notice, elapsed time and access to the incumbent of the first role prior to the incumbent of said first role having made said choice,

requiring that the incumbents of a third differentiated role have the opportunity to inspect the results of the choice made by the incumbent of the first role after said choice, and to accept or reject said results, with or without reference to established criteria, and

requiring that the incumbents of a fourth differentiated role have timely notice of the results of the choice made by the incumbent of the first role after said choice.

9. The method of claim 8, further comprising

requiring that the incumbents of a fifth differentiated role have the opportunity to inspect the results of the choice made by the incumbent of the first role after said choice, and to accept or reject said results according to its conformance or non-conformance to established criteria.

10. The method of claim 1, further comprising

using said process models to instantiate project models, and

using said process and project models to manage, direct, and control the work of the process.

11. The method of claim 2 further comprising

providing an abstract rule class as a subclass of the data class,

providing that said abstract rule class is specialized into concrete classes that include at least a class each of whose instances completely determine the result by choosing the value of its associated decision's data object, and

providing none or more additional concrete rule classes whose instances (i) determine the associated decision objects' requirement for some other specific data object, (ii) determine the associated decision objects' association with a specific role object, (iii) determine the incumbent of a specific role object associated decision's data object, and/or (iv) determine the use of a different role object associated with said decision object.

12. A computer implemented method for instantiating project models as instances of a process model to which they conform comprising

providing an extensible, object-oriented framework for modeling processes, and

providing abstract and concrete classes as elements of said framework, whose objects map plural participants in the process.

13. A computer implemented method for traversing networks including nodes and directed arcs connecting said nodes comprising

initializing all direct arcs and arc collections with an inactive state,

activating an entry collection of directed arcs which share a common entry node upon completion of the entry node's function,

activating all members of said entry collection upon activation of said entry collection,

activating an exit collection of directed arcs which share a common exit node upon activation of any member of said exit collection, and

testing, upon activation of said exit collection, other members of said exit collection for said member's active/inactive state and if any member of said exit collection is inactive, then stop testing and return said exit collection to its inactive state, and otherwise, if all members have tested active, activate their common exit node.